Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Critical Analysis of an Image Steganography Algorithm

[1]Surbhi Singhal*, [2]Rajkumar Singh Rathore, [3]Neeraj Rai
[1, 2] Department of Computer Science & Engg., Galgotia's College of Engg. and Technology, Greater Noida, U.P., India
[3]Department of Automation & Robotics, Ajay Kumar Garg Engineering College, Ghaziabad, U.P., India

Abstract— In the last few years Internet has developed manifold. Since, the development of Internet the most important and critical issue in information technology and communication field is the security of the information. Many methods like cryptography and others have been developed for securing the communication process and the information but it is not enough to keep only the contents of a message secret, it may also be necessary to keep the existence of the message secret. Steganography is the techniques which facilitates the said task. It is basically the science of invisible communication. There are different existing steganography techniques. This paper proposes a very efficient and offensive algorithm for Image Steganography which allows hiding message with in a 24-bit color image and as well as extract it. The proposed technique is implemented in MATLAB 7.5 and analyzed based on the PSNR and MSE value for different images. It is found that the proposed technique is very efficient having very low MSE value and very high PSNR value. Though the proposed technique is better, it still has the scope for improvement. In future we can combine the proposed algorithm with some cryptographic technique in-order to make it more powerful.

Keywords— Information hiding, histogram analysis, image steganography, peak signal to noise ratio, mean squared error, cover image, stego image.

I. INTRODUCTION

Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected [1]. Many simple steganographic techniques are being used from hundreds of years [2] [3], but the increasing use of data and information in an electronic format calls up for some new technique to be developed which can make it possible to hide information while communicating electronically. Steganography based on images [4] is one such technique which can facilitate the secure exchange of data electronically. Basically there are two important principles over which the computer steganography works: the first principle is that the digitized images or sound can be altered to a certain extent without causing any noticeable effect on them so as to hide the data in them; second principle deals with the human inability to distinguish minor changes in image color or sound quality, with which one can easily hide the data, be it 16 bit sound, 8 bit or even better 24-bit image. Speaking of image, changing the value of first component of pixel of an image wouldn't result in any detectable change of that color.

Steganography has been used throughout history and now-a-days with the advancement in technology it is being used widely [5] [6]. Throughout history, people have hidden information by a multitude of various methods and variations e.g. around 440 B.C. Histiaeus shaved the head of his most trusted slave and tattooed it with a message, which disappeared after the hair had regrown [7]. The purpose of this message was to instigate a revolt against the Persians. Another slave could be used to send a reply. During the revolution in America both the British and Americans used invisible ink which would glow over a flame to communicate secretly [8]. Steganography was also used in both World Wars. In First World War, prisoners of war would hide Morse code messages in letters home by using the dots and dashes on i, j, t and f. During Second World War, Germans would hide data as microdots. This involved photographing the message to be hidden and reducing the size so that it could be used as a period within another document. The director of FBI J. Edgar Hoover described the use of microdots as "the enemy's masterpiece of espionage" [6]. A message sent by a German spy during World War II read: "Apparently neutral's protest is thoroughly discounted and ignored isman hard hit blockade issue affects for pretext embargo on by-products, ejecting suits and vegetable oils." By taking the second letter of every word the hidden message "Pershing sails for NY June 1" can be retrieved [9].

Document layout can also be used to reveal information [10]. They can be marked and identified by modulating the positions of lines and words. With very discovery of a message hidden with an existing application, a new steganographic application is being devised. Old methods are given new twists. While drawings have often been used to conceal or reveal information, computer technology has sparked a revolution in such methods for hiding information.

Now-a-days steganography is based on the idea of hiding information and as computers are modern information machines, modern steganography is largely based on computer software techniques. Today, digital images as well as audio and video files offer a rich environment for hiding virtually unlimited types of data. Modern Steganographic tools are software programs with varying levels of sophistication that allow one to hide information in variety of places,

including filenames, unused disk space, network transmissions and executable code etc. This paper proposes and critically analyses an algorithm for steganography, which is based on difference scheme of RGB channels and text to be hidden. It uses histogram analysis which improves the quality parameters like Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE).

II. RELATED TERMINOLOGIES

A. RGB Channel

Channel is a traditional term which usually refers to some specific components of an image. It is always possible to extract the colour channel from any image irrespective to their stored format. There are three main channel types which are also called as different colour models namely RGB, CMYK and HSV. All have their own strengths and weaknesses. RGB has basically three individual channels: Red (R), Green (G) and Blue (B). It is mainly of 24-bits in which each channel has 8-bits.

B. Histogram Analysis

In context of image processing histogram primarily refers to the histogram drawn against the values of individual pixels intensity. It is basically the graph drawn for any particular image showing the number of pixels at different intensity values. For a color image histogram there are two possibilities, one is to consider the individual histogram of red, green and blue channels and other is to have a 3-D histogram taking all three channels together. It is very simple to plot the histogram, the image is scanned and the count of pixels at each intensity value is kept which is used to actually plot the histogram. The histogram of red, green and blue channel used in this algorithm is shown in fig.1, fig.2 and fig.3 respectively.

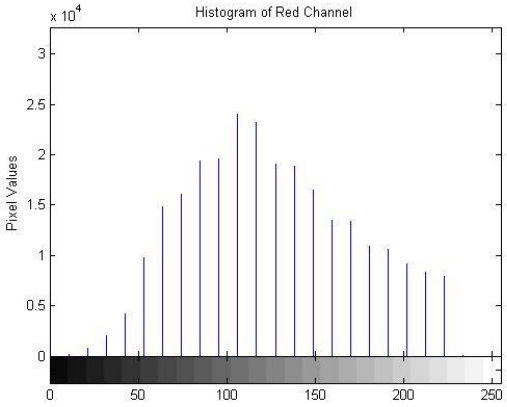

Fig.1: Histogram of Red Channel

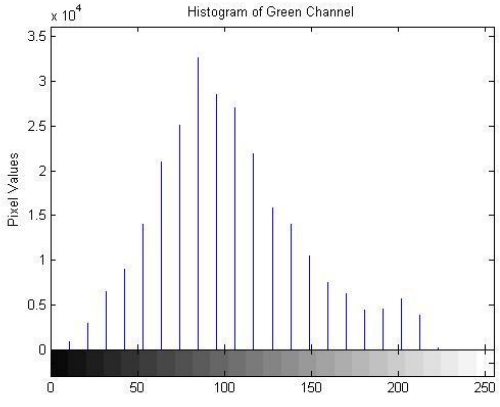

Fig.2: Histogram of Green Channel

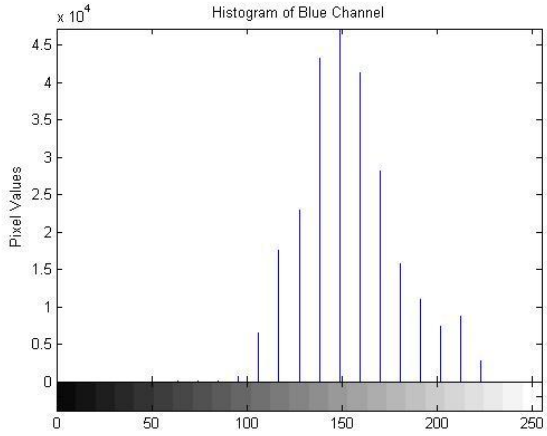

Fig.3: Histogram of Blue Channel

C. Mean Square Error (MSE)

Out of the two available error metrics used to compare the various images, MSE is one of that. It is basically the cumulative squared error between the original image and the obtained image. MSE is a risk function. It is the second moment of the error. Since last 5 decades MSE has been the most important quantitative performance metric in the field of signal processing. It mainly deals with the signal fidelity measure whose goal is to compare two signals and describe the degree of similarity or the level of error or distortion between them. In the domain of image processing this MSE is often converted into PSNR i.e peak-signal-to-noise-ratio measure.

The mathematical formulae to calculate the MSE in case of image processing is as follows:

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[I(x,y) - I'(x,y)]^2$$

Here, M, N are the dimensions of the image; I(x,y) is the original image and I'(x,y) is the distorted image. A lower value of MSE indicates that there is lesser error.

D. Peak Signal to Noise Ratio (PSNR)

It is an engineering term that measures the quality of the image and image reliability or conformity. It measures the ratio between the maximal possible power of the signal and the power of the corrupting noise.

Mathematically it is defined as,

$$PSNR = 20\ log_{10}\ (\frac{MAX_I}{\sqrt{MSE}})$$

Here, MAX_I is the maximum possible pixel value of the original image and MSE is the mean square error.

In PSNR 'signal' is the original image and 'noise' is the error in the stego image resulting due to encoding and decoding. It is basically a number that reflects the quality of the stego image and is measured in decibel (dB). Mathematically, PSNR is inversely proportional to MSE, which implies lower the value of MSE higher will be the PSNR value. Thus higher PSNR shows the better quality of the stego image.

III. PROPOSED WORK

The complete algorithm has been divided into two phases, one is embedding phase and the other is extracting phase:

A. Embedding of Secret Message in Cover Image

The proposed new steganography technique embeds the data in an image. Firstly, the second channel is taken which is used as the pixel indicator. It's every pixel's LSBs are set to zero. Now we start traversing each row sequentially of selected channel matrix and note the difference of the first character of the secret message with the first pixel of this channel till the difference falls in the range of -64 to 64 then this difference is noted and the corresponding second channel pixel LSB is set to one. Similarly, we move forward sequentially to note difference between the selected channel pixels and the characters of the text to be embedded.

The difference noted first time is embedded into the first six pixels LSBs of the third channel, the second difference is noted into the next six pixels LSBs of the third channel and so on. Thus, the message is embedded into the cover image by embedding the difference between the pixel of a selected channel and the character of the secret text message into this channel. This text embedded image is called the stego image.

This whole technique is developed in MATLAB R2007b.

Embedding Phase:

We summarize the process step-by-step as follows.

Algorithm 1. (The embedding process)

Input: A cover-image of w × h and secret data.

Output: A stego-image of w × h.

Step 1: Select a cover-image of w × h and secret data for hiding.

Step 2: Compute the histogram corresponding to the secret data with R, G and B plane of cover image.

Step 3: Initialize the LSBs of planes as zero say channel c_i;c_j, except the one say channel c_k which corresponds to the maximum probability of matching with secret data.

Step 4: Compare stego key and secret data with channel c_k sequentially.

Step 5: At matched point embed the difference with LSBs of channel c_i and a '1' on the LSB of channel c_j.

Data is embedded using above algorithm and stego image is produced and sent to the receiver or legitimate user. The receiver uses the shared stego key to extract secret message from stego image sent by sender to him. The process of extracting the data is discussed in the next section.

B. Extraction of Secret Message from Stego Image

Extraction of the secret message from the image can be done in the reverse way of the same technique, but the key must be the same as used in embedding the secret message into the images i.e. sender and receiver should know common stego key. If the key is not same, it is not possible for the receiver to decode the secret message from the stego image. Thus, this feature of matching key at the receiver side makes this technique a bit more secure in terms of attacks. The information of the histogram analysis is also transmitted from the sender to the receiver i.e. the receiver must know which channel are used for embedding difference, and pixel indicator.

Here, firstly we separate the three R G B channel from the stego image. After the stego key is matched, the second channel which is used as the pixel indicator channel is traversed sequentially and each pixels LSBs is observed, if the LSB of the pixel observed found to be 1, it's corresponding first channel pixel value is noted down and at the same time the difference value stored in the first six pixels LSBs of the third channel is recorded, now the difference of these two values gives the secret text character of the message. Similarly the process is repeated to extract the whole secret message.

Extracting Phase:

The extracting process is just reverse of the embedding process, which was explained in section 3.2. We, summarize the process step-by-step as follows.

Algorithm 2. (The extracting process)

Input: A stego-image of w × h.

Output: A secret data.

Step 1: Separate the c_i;c_j; c_k channel from the stego image.

Step 2: Scan for '1' in LSB of channel c_j.

Step 3: At matched point put corresponding values of channel c_k in array A[i] and difference from LSB of channel c_i in array B[i].

Step 4: Retrieve the secret message from the data array A[i] and difference array B[i]

IV. IMPLEMENTATION, RESULT & ANALYSIS

The proposed algorithm is implemented over the USC-SIPI (http://sipi.usc.edu/database/) image database which is the collection of digitized images which is primarily maintained to support the research in image processing. The overall database is divided into different volumes based on some basic characteristics of the images. The images in each volume are of different sizes like 256X256 pixels, 512X512 pixels, or 1024X1024 pixels. Out of all available volumes we have considered the miscellaneous volume from which we have taken 10 different images of size 512X512 pixels for obtaining the result. And we have considered four different size 40bits, 80bits, 104bits and 168 bits of message that is to be hidden, they are:

Message 1: S U R V I

Message 2: S U R V I M T e c h

Message 3: S U R V I M T e c h C S E

Message 4: S U R V I M T e c h C S E G a l g o t I a

The result obtained for one of the image (cov1 of size 512 X 512) is shown below:

Fig.4: Cover Image cov1

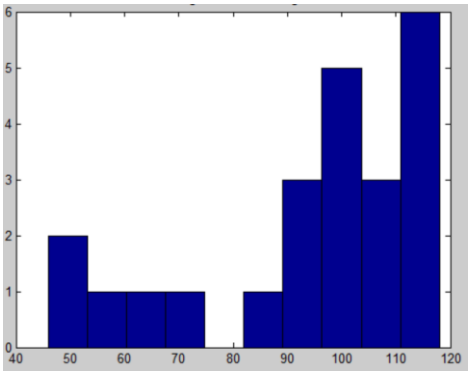

Fig.5: Histogram of Cover image cov1

Fig. 6: Stego Image of cov1

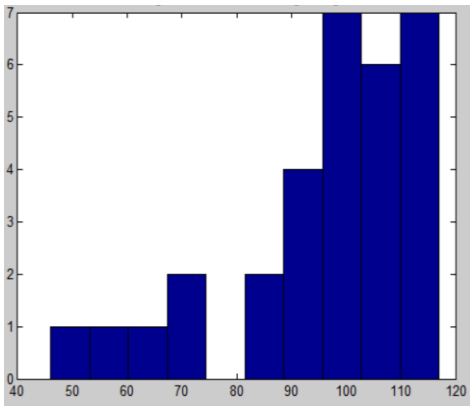

Fig.7: Histogram of Stego Image of cov1

Table 1: MSE & PSNR value for image cov1

	40bits	**80bits**	**104bits**	**168bits**
MSE	1.1444e-005	2.5431e-005	3.6875e-005	6.4850e-005
PSNR	108.3708	104.9029	103.2892	100.8375

Similarly, the result has been obtained for different sets of images. The obtained results have been plotted in different ways in-order to analyze the algorithm.

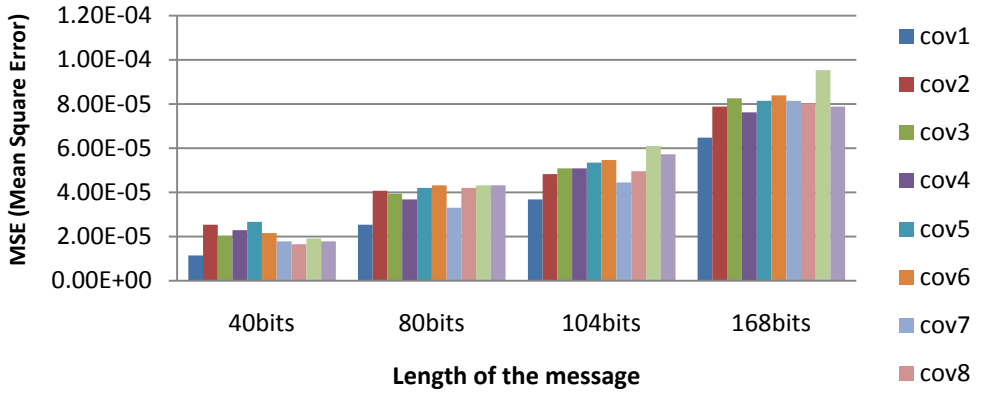

Fig. 8: Variation of MSE with respect to message length for different images

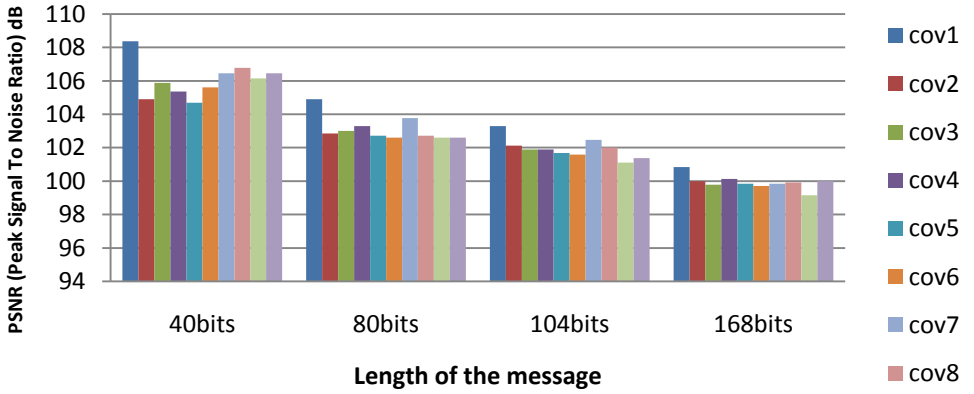

Fig.9: Variation of PSNR with respect to message length for different images

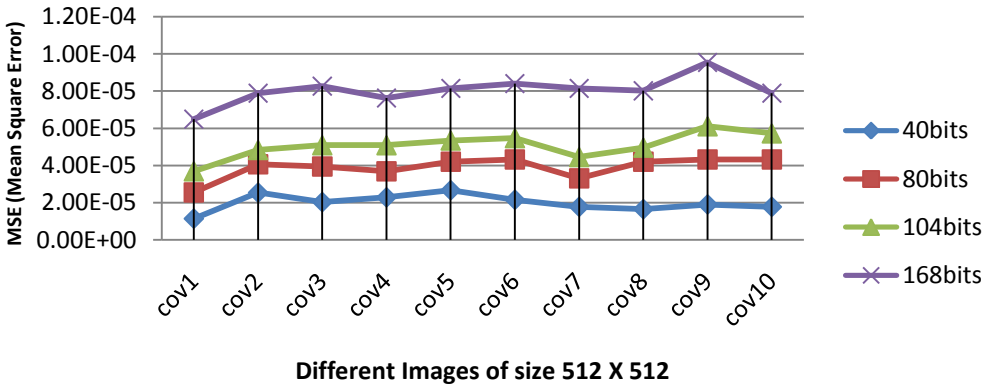

Fig. 10: MSE values for different images

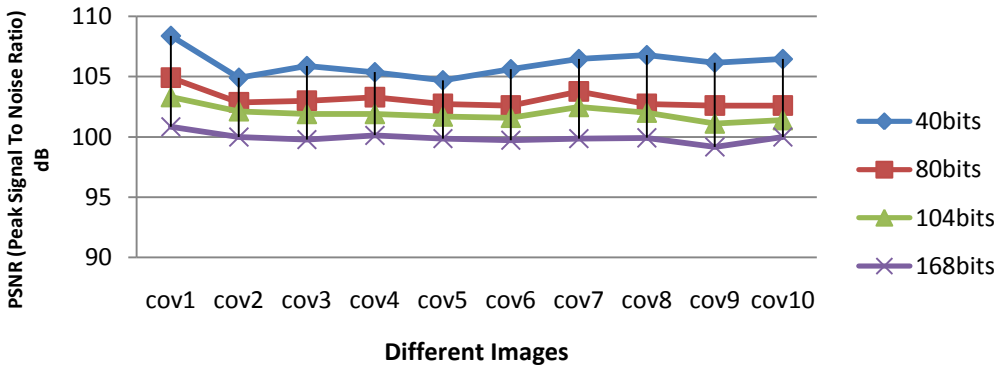

Fig. 11: PSNR values for different images

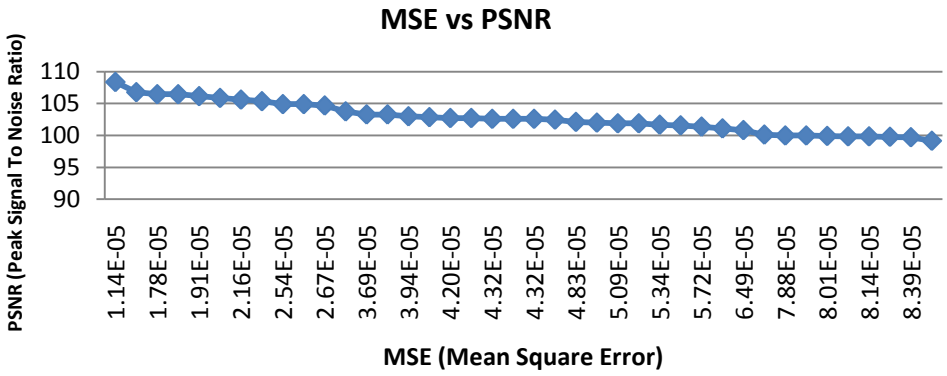

Fig. 12: MSE Vs PSNR

Fig.8 shows that the MSE value increases with increase in size of the message. Fig.9 shows that the PSNR value decreases with increase in size of the message. Fig.10 shows that the MSE is very low as well is almost same for every image which proves the correctness of the proposed algorithm. Fig. 11 shows that PSNR value is very high as well as almost same for different images which prove the correctness of the proposed algorithm. Fig.12 shows that MSE and PSNR are inversely proportional with each other.

The overall result analysis shows that if the amount of data embedded in the image is increased then also there is a minor change in the values of MSE and PSNR as shown in fig.8 and fig.9. The proposed algorithm has improved the value of MSE tremendously as it lies in the span of 0 to 1. The PSNR value is higher from all the previous steganography techniques, which shows good quality of the stego-images that is highly acceptable by the human eyes.

V. CONCLUSION

In this paper we have proposed and analyzed an algorithm for image steganography. The proposed algorithm has two parts, the first part embeds the information and the second part extracts the hidden information. The algorithm has been implemented in MATLAB 7.5 for different example images. One example is presented in this paper. For the purpose of the quality, we have calculated the PSNR value, for the presented example as shown in table 1. The proposed algorithm has improved the value of MSE tremendously as it lies in the span of 0 to 1. The PSNR value is higher from all the previous steganography techniques, which shows good quality of the stego-images that is highly acceptable by the human eyes.

REFERENCES

[1] Provos, Niels, and Peter Honeyman. "Hide and seek: An introduction to steganography." Security & Privacy, IEEE 1.3 (2003): 32-44.

[2] Katzenbeisser, Stefan, and Fabien Petitcolas. Information hiding techniques for steganography and digital watermarking. Artech house, 2000.

[3] Morkel, Tayana, Jan HP Eloff, and Martin S. Olivier. "An overview of image steganography." ISSA. 2005.

[4] Hamid, Nagham, et al. "Image steganography techniques: an overview."International Journal of Computer Science and Security (IJCSS) 6.3 (2012): 168-187.

[5] Kahn, David. "The history of steganography." Information Hiding. Springer Berlin Heidelberg, 1996.

[6] Kumar, Arvind, and Km Pooja. "Steganography-A data hiding technique."International Journal of Computer Applications 9.7 (2010): 19-23.

[7] Jamil, Tariq. "Steganography: the art of hiding information in plain sight."Potentials, IEEE 18.1 (1999): 10-12.

[8] Yusof, Yusnita, and Othman O. Khalifa. "Digital watermarking for digital images using wavelet transform." Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on. IEEE, 2007.

[9] Singh, Kh Manglem, et al. "Hiding secret message in edges of the image."Information and Communication Technology, 2007. ICICT'07. International Conference on. IEEE, 2007.

[10] Johnson, Neil F., and Sushil Jajodia. "Exploring steganography: Seeing the unseen." Computer 31.2 (1998): 26-34.

